Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Autoencoder-bank Based Design for Adaptive Channel-Blind Robust Transmission

Hossein Safi
Shahid Beheshti University

Mohammad Akbari (m.akbari@itrc.ac.ir)
ICT Research Institute https://orcid.org/0000-0002-9339-1201

Elaheh Vaezpour
ICT Research Institute

Saeedeh Parsaeefard
University of Toronto

Raed M. Shubair
Massachusetts Institute of Technology

Research

Keywords: deep learning, autoencoder, blind estimation

Posted Date: September 16th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-72829/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Version of Record: A version of this preprint was published on March 6th, 2021. See the published version at https://doi.org/10.1186/s13638-021-01929-z.

RESEARCH

Autoencoder-Bank Based Design for Adaptive Channel-Blind Robust Transmission

Hossein Safi[1], Mohammad Akbari[2*], Elaheh Vaezpour[2], Saeedeh Parsaeefard[3] and Raed M Shubair[4]

[*]Correspondence: m.akbari@itrc.ac.ir, Iran Telecom Research Center
[1]Department of Electrical Engineering, Shahid Beheshti University, Velenjak, Tehran, Iran
Full list of author information is available at the end of the article

Abstract

The idea of employing deep autoencoders (AEs) has been recently proposed to capture the end-to-end performance in the physical layer of communication systems. However, most of the current methods for applying AEs are developed based on the assumption that there is an explicit channel model for training that matches the actual channel model in the online transmission. Since the actual channel varies over time, this imposes a major limitation on employing AE-based systems. In this paper, without relying on an explicit channel model, we propose an adaptive scheme to increase the reliability of an AE-based communication system over different channel conditions. More precisely, we divide the interval of random channel coefficients into n sub-intervals. Subsequently, in the offline training phase, we employ an AE bank consisting of n pairs of encoder and decoder and perform training over the sub-intervals. Then, in the online transmission phase, based on the actual channel conditions, the optimal pair of encoder and decoder is selected for data transmission in terms of satisfying an average block error rate (BLER) constraint imposed on the system. To monitor actual channel conditions for adopting the adaptive scheme, we assume a realistic scenario where the instantaneous channel gain is not known to Tx/Rx and it is blindly estimated at the RX, i.e., without using any pilot symbols. Our simulation results confirms the superiority of the proposed adaptive scheme over a non-adaptive scenario in terms of average power consumption. For instance, when the target average BLER is equal to 10^{-4}, our proposed algorithm with $n = 5$ can achieve a performance gain over 1.2 dB compared with a non-adaptive scheme.

Keywords: deep learning; autoencoder; blind estimation

1 Introduction

To reliably transmit data from a source to a destination, conventional communication systems employ multiple independent blocks which are separately optimized to perform isolated functions (e.g., source/channel coding, modulation, channel estimation, equalization) [1]. However, such a divided architecture is known to be sub-optimal [2], and thus, achieving optimal performance through the end-to-end optimization of a communication system retains appeal for carrying out further investigations [3].

Recently, considerable advances in Deep Learning (DL) neural networks (NNs) have empowered us to efficiently perform end-to-end learning of communication systems [3]. This way, transmitter (Tx) and receiver (Rx) can be trained in an end-to-end fashion under a specific performance metric and channel model [4]. Accordingly, via modeling the Tx and Rx as NNs, autoencoders (AEs) have emerged

as a useful tool for end-to-end modeling of the physical layer of communication systems [5]. In particular, such a setup is enable to optimize Tx and Rx without being limited to conventional component-wise optimization methods, and hence, moving away from carefully optimized sub-blocks to adaptive and flexible NNs [6]. Using offline data-set, AEs can be trained and optimized for a practical communication system, and this architecture can outperform the conventional separable design of the physical layer of such systems.

Owing to these benefits, a number of studies on employing AEs in the physical layer of communication systems has been reported [6, 7, 8, 9, 10]. Particularly, the idea of end-to-end learning of communication systems through deep NN-based AEs has been applied to a communication system characterized by an orthogonal frequency division multiplexing with cyclic prefix in [6]. Moreover, the authors in [7] investigates the problem of joint source and channel coding of structured data (i.e, natural language) over a noisy channel and attains lower word error rates by developing an AE-based system. A new AE-based peak-to-average power ratio reduction scheme has been proposed in [8]. Furthermore, the authors in [9] develop an AE-based deep learning architecture to model a multiuser single-input multiple-output communication system. The work in [10] employs an AE to find proper constellations and corresponding receiver devices when a radar systems coexists with interfering wireless systems.

Nevertheless, most of these prior works assume an exact mathematical channel model to perform training. More precisely, in an end-to-end communication system, the channel is considered as a layer in the NN. Thus, to backpropagate error during the training phase, the AE needs to know the gradient of the channel transfer function. Moreover, to capture the maximum end-to-end performance, considered channel model in the training phase must match the actual channel model in the online transmission phase. This imposes a major limitation on employing AE-based approaches to achieve maximum end-to-end performance of a communication system when the actual channel varies over the time. To cope with this problem, prior works have proposed different online training methods based on measured data during online transmission. For instance, the work in [2] considers training the DL-based system using a channel model, and then fine-tuning the Rx with measured data. However, fully capturing of end-to-end performance is not possible since no fine-tuning is done at the Tx side in this approach. Moreover, the authors in [11] approximate the loss function gradient with respect to the Tx parameters and develop an alternating algorithm for end-to-end training without channel model knowledge. This algorithm iterates between two phases: (i) training of the Rx using the true gradient of the loss, and (ii) training of the Tx based on an approximation of the loss function gradient. However, this method takes more samples to converge and is relied on a two phase time-consuming training paradigm over online transmission, thus decreasing the link availability.

To mitigate the need for undergoing complex online training over actual channels as well as to obtain the maximum end-to-end performance of a communication system, we propose a robust adaptive scheme for data transmission over a random

channel with no specific mathematical model using an AE bank[1]. More precisely, we divide the interval of random channel coefficients into n sub-intervals. Subsequently, in the offline training phase, we employ n pairs of AE consisting of encoder, channel, and decoder, each of which corresponds to a specific sub-interval. For online transmission, we assume a realistic scenario where the instantaneous channel gain is not known to Tx/Rx. Thus, we need to estimate the channel gain at the RX and feed it back to the Tx. Then, based on the actual channel conditions in the online transmission phase, the pair of encoder and decoder that satisfies the system average block error rate (BLER) constraint is selected for data transmission. To increase bandwidth efficiency, as well as to avoid data framing at the Tx, we propose a method to estimate the channel blindly, i.e., without using any pilot symbols. We then compare the proposed blind method for channel estimation with a pilot-based method in terms of average BLER of the system. Moreover, the performance of the proposed adaptive scheme is evaluated in terms of the required number of encoders and decoders and also the average power consumption to satisfy a BLER constraint for data transmission. In this regard, we seek to balance an inherent tradeoff between the deployment cost (represented by the required number of encoders and decoders), and the system performance (represented by the average power consumption and target BLER). Simulation results show the effectiveness of our proposed adaptive method in terms of average BLER over different channel conditions compared to a non-adaptive scheme.

The rest of this paper is organized as follows. Section 3 describes our system model including the end-to-end communication system and the channel estimation methods. In Section 4, we present our adaptive transmission scheme. In Section 5, we present the numerical results of the proposed adaptive scheme and system performances. Finally, we conclude the paper in Section 6.

2 Methods

The inspiration for an end-to-end deep learning model, also known as, autoencoder, is rooted in the functioning of the proposed method in this study. More precisely, Figure 1 represents the layered structure of the end-to-end deep learning-based communication system modeled as an autoencoder. Accordingly, without relying on an explicit channel model, we propose an adaptive scheme to increase the reliability of an AE-based communication system over different channel conditions. To model and simulate the proposed design as well as to build and train the deep neural network, we use Keras [12] with TensorFlow [13] in its back-end. Moreover, the set of parameters for simulation is provided in TABLE 1. During the training process, the input symbols, modeled as one-hot vectors, go through the autoencoder where the weights of the neural nodes are initialized with random values. After that, the weight vectors of the nodes will be tuned. The main training goal is to minimize the loss function and maximize the accuracy of the whole process.

[1]In this paper, the term "robust" indicates that, by using the proposed adaptive scheme, the performance of the system will not be affected by channel variations over the time.

3 System Model

3.1 End-to-End Communication System (Autoencoder)

We assume a DL-based communication system modeled as an AE. Particularly, the AE includes three main blocks, namely, the Tx, the Rx, and channel, as shown in Fig. 1,. The input message $s \in \mathcal{M} = \{1, 2, ..., M\}$, has been received by the Tx. Here, we have $M = 2^b$ where b is the number of bits per message. Before entering to the dense layers, the input message is transformed into a one-hot vector $\mathbf{s}_m$ of dimension M which consists of a single element equal to "1" in position m, whereas all other elements are equal to "0". After passing the one-hot vector through the multiple dense layers at the Tx, the transmitted signal $\mathbf{x} = [x[1], \ldots, x[L]]$ is formed for L discrete channel uses. Accordingly, for the considered setup, the data rate is defined as $r = \frac{b}{L}$ (bit/channel use). Furthermore, the Tx last layer normalizes the transmit vectors to guarantee that the average energy per symbol is equal to a predefined values $E_s = \left[\frac{1}{L}\|\mathbf{x}\|_2^2\right]$, where $\|\cdot\|_2$ is the Euclidean norm.

The channel is implemented by including both fading and noise layers whose output $\mathbf{y} = [y[1], \ldots, y[L]]$, i.e., a noisy and distorted version of $\mathbf{x}$, is given by

$$\mathbf{y} = h\mathbf{x} + \mathbf{w} \tag{1}$$

where h is the channel gain[2] , and $\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \sigma_L^2 \mathbf{I}_L\right)$ is a zero-mean additive white Gaussian noise (AWGN) vector. At the Rx side, the received signal $\mathbf{y}$ is passed through multiple dense layers to reach the last layer. Accordingly, at this layer, a softmax activation function is utilized where its output consists of an estimate of the corresponding posterior probability vector $\mathbf{p} \in \mathbb{R}^M$ over all possible messages. The index of the element of $\mathbf{p}$ with the largest value is returned to estimate the transmitted message $\hat{s}$ based on the maximum a posteriori (MAP) criterion. Moreover, we utilize the mean square error (MSE) as the loss function of the training process. This way, the loss function is obtained as

$$\mathcal{L}(\mathbf{s}_m, \mathbf{p}) = \|(\mathbf{s}_m - \mathbf{p})\|_2^2 . \tag{2}$$

Also, the BLER of considered setup is obtained as

$$P_e = \sum_{s \in \mathcal{M}} \Pr(s \neq \hat{s}) \mathrm{p_s} \tag{3}$$

where $\mathrm{p_s}$ is the *a priori* probability of transmission message s. Since the message probability distribution is commonly assumed to be uniform, we have $\mathrm{p_s} = \frac{1}{M}$.

3.2 Challenge in Training Over Physical Channel

To fully exploit the end-to-end performance of an AE-based communication system, channel model in the training phase must match the actual channel over which the system is supposed to communicate. Nevertheless, for an actual system, the channel

[2]It is worth mentioning that, the NN architecture is not able to perform complex operation and complex numbers are represented by two real numbers. As a result, all channel gains are real-valued.

is not perfectly known and varies over the time. Therefore, an AE that is trained over a specific realization of the channel may not deliver the expected performance with changing in channel conditions. Thus, AE should be re-trained from scratch for each new channel condition in order to minimize the loss function and BLER which is a time-consuming process that greatly restricts the link availability and reliability. In this paper, we propose to obviate this practical limitation by employing an autoencoder bank consisting of multiple pairs of trained encoder and decoder for different channel conditions. Subsequently, regarding the actual channel state in the online transmission phase, one pair of trained encoder and decoder is selected for data transmission. To this aim, we estimate the channel gain at the Rx and feed it back to the Tx. The details of the proposed adaptive scheme are provided in Section 4. To monitor actual channel conditions for adopting the adaptive scheme, and, depending on whether pilot symbols are used or not, two methods for estimating the channel gain of the considered AE-based communication system are proposed in the sequel.

3.2.1 Channel Estimation Using Pilot Symbols

We assume that the pilot symbol s' is transmitted. Therefore, the channel output $\mathbf{y}'$ corresponding to the encoded signal $\mathbf{x}'$ is given by

$$\mathbf{y}' = h\mathbf{x}' + \mathbf{w}. \tag{4}$$

Given $\mathbf{x}'$, one can obtain an estimate of the channel gain, h, by applying the maximum-likelihood (ML) criterion as

$$\begin{aligned} \hat{h}_{ML} &= \arg\max_{h} \; P(\mathbf{y}'|\mathbf{x}';h) \\ &= \arg\min_{h} \; |\mathbf{y}' - h\mathbf{x}'|^2 = \frac{\langle \mathbf{x}',\mathbf{y}' \rangle}{\langle \mathbf{x}',\mathbf{x}' \rangle} \end{aligned} \tag{5}$$

where $\langle \cdot,\cdot \rangle$ denotes the dot product. Although estimating the communication channel via pilot symbols results in an accurate estimation, adopting this approach requires data framing at the Tx side which leads to the loss of data rate and increases the required channel bandwidth. In the sequel, we propose a near optimal method for channel estimation without using pilot symbols.

3.2.2 Blind Channel Estimation

To increase the bandwidth efficiency by avoiding pilot symbols and, at the same time, to decrease the system complexity by avoiding data framing at the Tx, here, we tend to estimate the channel in a blind way, i.e., without using any pilot symbols. For this purpose, we assume that the encoded symbol is accumulated during an observation window composed of L intervals. As a result, at each interval k, $k \in \{1, \ldots, L\}$, the received signal (or equally the channel output) is obtained as

$$y\,[k] = hx\,[k] + w\,[k]\,. \tag{6}$$

Squaring both sides of (6) implies that

$$\begin{aligned} y\left[k\right]^2 &= \left(hx\left[k\right]+w\left[k\right]\right)^2 \\ &= h^2x\left[k\right]^2+2hx\left[k\right]w\left[k\right]+w\left[k\right]^2. \end{aligned} \tag{7}$$

Since the term $w\left[k\right]^2$ is much smaller than $w\left[k\right]$ i.e., $w\left[k\right]^2 \ll w\left[k\right]$, we can resonantly neglect $w\left[k\right]^2$ and approximate eq. 7 as

$$y\left[k\right]^2 \simeq h^2x\left[k\right]^2+2hx\left[k\right]w\left[k\right]. \tag{8}$$

By performing summation over k, eq. 8 can be reformulated as

$$\tau_y \simeq h^2+w'' \tag{9}$$

where $\tau_y = \frac{1}{LE_s}\sum_k y\left[k\right]^2$, and w'' is a zero-mean white Gaussian noise with variance $\sigma^2 = \dfrac{4h^2\sigma_N^2}{LE_s}$. Therefore, τ_y has a Guassian distribution with a mean equal to h^2 and variance σ^2. By using the ML criterion, a blind estimate of the channel is attained as

$$\begin{aligned} \hat{h}_{BL} &= \arg\min_h \left|\tau_y-h^2\right|^2 \\ &= \sqrt{\tau_y}. \end{aligned} \tag{10}$$

Moreover, by employing a buffer at the Rx, the communication system can support a real-time decision-feedback channel estimation process in which symbol-by-symbol estimations are carried out using previous estimations and an observation window of L received statistics.

4 Adaptive Transmission Scheme

In this section, an adaptive transmission scheme is proposed for the AE-based communication system to increase the system reliability over different channel conditions. Meanwhile, we aim to minimize the number of trained encoder-decoder pairs required for the communication link with the average BLER, and transmit power constraints under different channel conditions. Figure 2 depicts the four-step structure of our proposed adaptive DL-based transmission scheme. Each step is described as follows.

The first step is offline training. Different pairs of the encoder and decoder in the AE bank are trained offline over different channel conditions. Since the AE-represented communication system should be applicable for any type of channel without a tractable mathematical model, in this paper, we assume there is no channel model information. As a result, the channel fading block acts as a random gain block where its input is multiplied by a random number (i.e., the instantaneous channel gain) to produce the output[3] . We divide the interval of channel gains,

[3]It is worth noting that, generally, there is no tractable mathematical model in a real-world communication system. More precisely, in the context of AE, the actual

Algorithm 1 Robust adaptive DL-based end-to-end transmission using real-time channel estimation.

Input: n sub-intervals of the channel gain, n pairs of encoder and decoder;
Output: An estimate value for the actual communication channel $\hat{h}$, i^{th} pair of encoder and decoder for sending and receiving data;
Step 1 (offline): Train n pairs of encoder and decoder over n channel sub-intervals;
Step 2 (online): Estimate the instantaneous channel gain, $\hat{h}$, at the Rx using 10, and feed it back to the Tx;
Step 3 (online): if $\hat{h}$ lies in the ith sub-interval, choose the ith pair of encoder and decoder for sending and receiving data;
Step 4 (online): Monitor the values of real-time channel estimation and switch to the appropriate pair of encoder and decoder whenever the estimated value has changed from the current sub-interval to another one.

$\mathbf{h} = [h_{\min}\ h_{\max}]$, into n sub-intervals. Therefore, the channel gain interval is obtained as

$$\mathbf{h} = \underbrace{[[h_{\min}\ h_1], (h_1\ h_2], \ldots, (h_{\text{n-1}}\ h_{\max}]]}_{n \text{ sub-intervals}}. \tag{11}$$

Accordingly, in the training phase, we have n fading blocks for which the gain of each block is randomly selected from its associated sub-interval. Then, AE_i (the ith pair of encoder, and decoder for each sub-interval, $i \in \{1, \ldots, n\}$), is trained under the assumption that the channel gain in the channel layer of AE_i lies in $(h_i - 1, h_i]$. After training, the trained encoders and decoders are employed with fixed parameters (i.e, the input and output weights and bias of the neurons remain constant during online transmission). As a result, each trained pair is optimized for a specific channel condition and is used when the practical channel conditions are within the same interval as the one used in the training. Note that, by training the system for each sub-interval, the associated encoder carefully learns a robust representations $\mathbf{x}$ of the different symbol $\mathbf{s}$ regarding to the possible distortions created by the channel at that interval. Therefore, the whole system is expected to deliver a robust performance over a wide range of channel conditions.

The second step is channel estimation during online transmission. As we mentioned in Section 3.2, the Rx can perform channel estimation by employing two methods, i.e., using pilot symbols, and blind estimation. For estimating the channel via pilot symbol, the Tx should enclose transmit data in different frames and insert pilot symbols into the frames which results in less bandwidth efficiency. To avoid using pilot symbols, the Rx can also perform blind channel estimation over an observation window as thoroughly discussed in Section 3.2.2.

In the third and fourth steps, first, the estimated channel gain at the Rx, $\hat{h}$, feeds back to the Tx. Next, when $\hat{h}$ lies in the ith sub-interval, the ith pair of encoder and decoder is selected for sending and receiving data. We summarize the main steps of the proposed adaptive transmission scheme in Algorithm 1.

It should be noted that the robust performance of the proposed adaptive scheme comes at the expense of using multiple pairs of encoder and decoder instead of

channel is generally considered as a black box for which only the inputs and outputs can be observed. Here, we just need to perform some simple measurements at different time intervals to determine the range of the channel gain variations.

using one pair which increases the deployment cost. Clearly, this price should be paid for being agnostic to the actual channel during training phase. This gives rise to a natural question: what is the minimum number of pairs of encoder and decoder which satisfies a target level of performance for the considered system. In the sequel, we propose an answer to this question by evaluating the performance of the proposed adaptive scheme in terms of average power consumption and deployment cost to fulfill a predetermined average BLER. More precisely, we impose a target average BLER, P_e^t, as a constraint for the system, and minimize the number of encoder and decoder pairs (or equally the number of sub-intervals n), to satisfy the BLER constraint under different channel conditions. Hence, the optimization problem can be formulated as

$$\min \; n \tag{12a}$$

$$\text{s.t.} \quad \bar{P}_e \leq P_e^t, \tag{12b}$$

Note that, in the optimization problem 12, the minimum number of encoder and decoder pairs should be found under different channel conditions and average transmit power. From the system performance's perspective, the more encoders and decoders, the more robust is the system which is designed for different channel conditions. Therefore, finding the optimum number of encoders and decoders requires balancing an inherent tradeoff between the cost of system deployment and desirable performance. This tradeoff is thoroughly studied in Section 5.

5 Numerical Results and Discussion

In this section, we evaluate the numerical results of the proposed adaptive transmission scheme, in the DL-based communication system. We use Keras [12] with TensorFlow [13] in its back-end in order to build and train our deep NN. For training, we use a variant of SGD known as Adam with widely accepted thumb rule for the parameter values as follows, learning rate $\eta = 0.001$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$ [2]. Also, the set of parameters for each pair of encoder and decoder (the AE) is provided in TABLE 1.

We first compare the proposed blind method for channel estimation with pilot estimation in terms of average BLER. To this aim, we plot the average BLER of the considered system when different methods for channel estimation are employed in Fig. 3. The results of this figure are obtained by assuming $M = 16$ and $L = 7$, thus $r = \frac{4}{7}$. Moreover, in Fig. 3, we assume a communication system employing binary phase-shift keying (BPSK) modulation and a Hamming $(7, 4)$ code with either binary hard-decision decoding or soft decoding against the BLER achieved by the trained AEs as a baseline system for comparison. From the results of this figure, first, we can observe that given the same information transmission rate, $r = \frac{4}{7}$, the performance of the DL-based system is better than that of communication system employing Hamming code. It is worth mentioning that, the DL-based system does not employ any error control coding approach for the noisy channel, and it still outperforms a classical communication system that utilizes error control schemes. Second, for the considered rate in our setup and compared with the pilot estimation, the proposed blind method for channel estimation achieves an acceptable level of

accuracy (SNR gap less than 0.3 dB for $L = 7$ in our setup). Thus, proposed blind estimation method can be applied in our considered adaptive scheme. To investigate the tradeoff of finding the optimum number of encoder and decoder over different channel conditions, we have presented curves of average BLER as a function of p_t for different number of sub-intervals, n, and for the case of an AE with $M = 16$, and $L = 7$ in Fig. 4. Firstly, as expected, our considered system gives its worst performance in terms of average BLER when one pair of encoder and decoder is used for all channel conditions, i.e., when $n = 1$ (non-adaptive scheme). Indeed, this worst performance highlights the necessity of employing a robust transmission scheme over different channel conditions. Subsequently, by applying our adaptive scheme, one can observe that the performance of the considered system improves by increasing n. On the other hand, from a specific number onwards (e.g., $n = 8$ in our setup), increasing the number of sub-intervals (or equally the number of pairs of encoders and decoders), does not necessarily improve the system performance. This can be justified by the fact that, in this case, the n pairs of encoder and decoder are trained over closed sub-intervals (i.e., nearly the same channel conditions) as the number of these sub-intervals increases. Hence, when the sub-interval is short, those pairs of encoder and decoder trained over close sub-intervals deliver the same performance when they are used in actual channel conditions. Hence, the improvement of performance is negligible in this situation. Finally, from the results of Fig. 4, the minimum value of average transmit power, and also the minimum number of pairs of encoder and decoder can be obtained to satisfy the average BLER constraint in eq. (12). For instance, as we can observe form Fig. 4, for a target average BLER equal to 10^{-4}, our proposed algorithm with $n = 5$ can achieve a performance gain over 1.2 dB in terms of average power consumption compared with a non-adaptive scheme.

6 Conclusion

In this paper, we proposed an adaptive scheme to increase the reliability of an AE-based communication system over different channel conditions. Accordingly, we divided the interval of random channel gains into n sub-intervals and assigned n pairs of encoder and decoder to each interval and performed offline training. Subsequently, regarding the actual channel state in the online transmission phase, one pair of trained encoder and decoder is selected for data transmission. To this aim, we estimated the channel gain at the Rx and feed it back to the Tx without using any pilot symbols. We showed that, compared with a non-adaptive scheme, by using the proposed adaptive method, the DL-based system can deliver a robust performance in terms of average BLER over different channel conditions. Also, we showed that our proposed adaptive scheme can achieve a performance gain in terms of average power consumption to achieve the same average BLER as a non-adaptive system.

List of Abbreviations
AE: Autoencoder
AWGN: Additive white Gaussian noise
BLER: Block error rate
BPSK: Binary phase shift keying
DL: Deep learning
MAP: Maximum a posteriori
ML: Machine learning
Rx: Receiver
SGD: Stochastic gradiant descent
Tx: Transmiter

Competing interests
The authors declare that they have no competing interests.

Author's contributions
Hossein Safi, and Mohammad Akbari are the main authors of the current paper. They contributed to the development of the ideas, design of the study, theory, result analysis, and paper writing. Elahe Vaezpour, Saeedeh Parsaeefard, and Raed M Shubair contributed to the paper revision. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
Not applicable.

Author details
[1]Department of Electrical Engineering, Shahid Beheshti University, Velenjak, Tehran, Iran. [2]Iran Telecom Research Center, North Karegar, Tehran, Iran. [3]Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada. [4]Massachusetts Institute of Technology (MIT), Massachusetts, CA 02139, USA & New York University (NYU) Abu Dhabi, Abu Dhabi 129188, UAE,.

References
1. Goldsmith, A.: Wireless Communications. Cambridge University Press, ??? (2005)
2. Dörner, S., Cammerer, S., Hoydis, J., ten Brink, S.: Deep learning based communication over the air. IEEE J. Sel. Topics Signal Process. **12**(1), 132–143 (Dec. 2017)
3. O'Shea, T., Hoydis, J.: An introduction to deep learning for the physical layer. IEEE Trans. Cogn. Commun. Netw. **3**(4), 563–575 (Dec. 2017)
4. Qin, Z., Ye, H., Li, G.Y., Juang, B.-H.F.: Deep learning in physical layer communications. IEEE Wireless Commun. **26**(2), 93–99 (Mar. 2019)
5. O'Shea, T.J., Karra, K., Clancy, T.C.: Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention. (Dec. 2016). in *Proc. IEEE 16th International Symposium on Signal Processing and Information Technology (ISSPIT)*, Limassol, Cyprus
6. Felix, A., Cammerer, S., Dörner, S., Hoydis, J., Ten Brink, S.: OFDM-Autoencoder for end-to-end learning of communications systems. (2018). in *Proc. IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Kalamata, Greece
7. Farsad, N., Rao, M., Goldsmith, A.: Deep learning for joint source-channel coding of text. (Apr. 2018). in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Alberta, Canada
8. Kim, M., Lee, W., Cho, D.-H.: A novel PAPR reduction scheme for OFDM system based on deep learning. IEEE Commun. Lett. **22**(3), 510–513 (Mar. 2017)
9. Xue, S., Ma, Y., Yi, N., Tafazolli, R.: Unsupervised deep learning for MU-SIMO joint transmitter and noncoherent receiver design. IEEE Wireless Commun. Lett. **8**(1), 177–180 (Aug. 2018)
10. Alberge, F.: Deep learning constellation design for the AWGN channel with additive radar interference. IEEE Trans. Commun. **67**(2), 1413–1423 (Oct. 2018)
11. Aoudia, F.A., Hoydis, J.: End-to-end learning of communications systems without a channel model. (Oct. 2018). in *Proc. IEEE 52nd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA
12. Chollet, F., et al.: Keras. https://keras.io (2015)
13. Abadi, M., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org (2015). http://tensorflow.org/

Figures

Figure 1 The layered structure of an end-to-end deep learning-based communications system modeled as an autoencoder.

Figure 2 Four-step structure of the proposed adaptive transmission scheme for the DL-based communication system.

Figure 3 Average BLER versus SNR when the instantaneous channel gain, $\hat{h}$, is estimated by using two methods, i.e., pilot symbol method, and blind estimation method. The parameters for the AE are set as follows: $M = 16$, $L = 7$, and $n = 5$.

Figure 4 Average BLER versus p_t for different numbers of sub-intervals n. The parameters for the AE are set as follows: $M = 16$, $L = 7$.

Tables

Table 1 Layout of Autoencoder

	Parameter #	Output shape
Tx:		
Input	0	one-hot $\in \mathbb{R}^M$
Dense (Relu)	72	$\mathbb{R}^M$
Dense (Relu)	72	$\mathbb{R}^M$
Dence(linear)	63	$\mathbb{R}^L$
Normalization	14	$\mathbb{R}^L$
Channel:		
Random gain	14 (non-trainable)	$\mathbb{R}^L$
AWGN	14 (non-trainable)	$\mathbb{R}^L$
Rx:		
Dence(Relu)	64	$\mathbb{R}^M$
Dence(Softmax)	72	$\mathbb{R}^M$

Figures

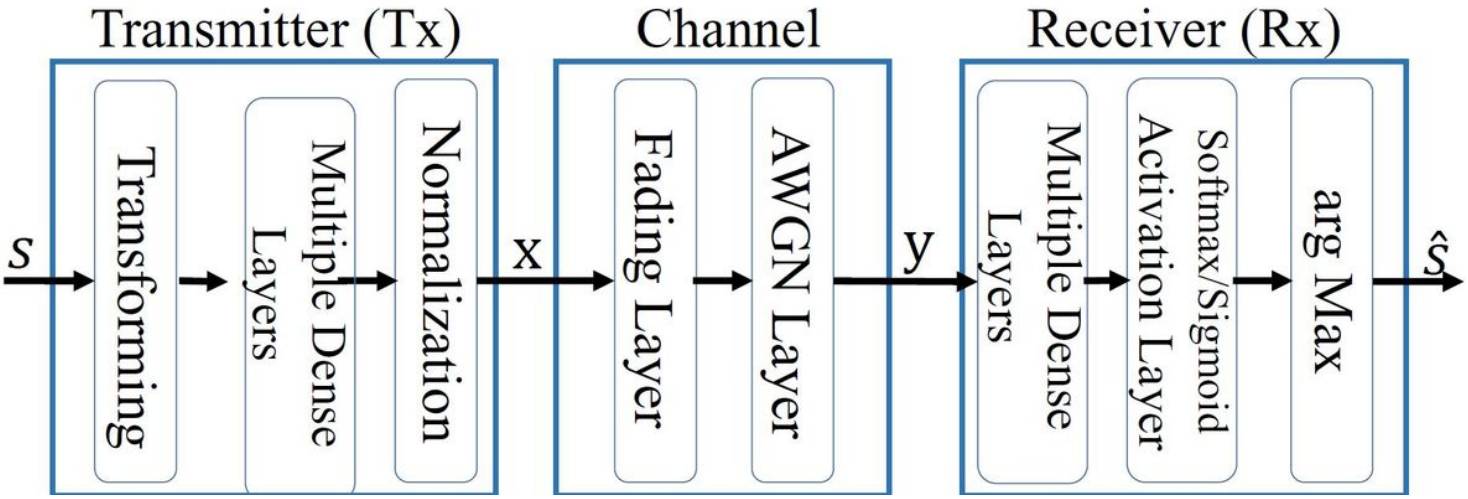

Figure 1

The layered structure of an end-to-end deep learning-based communications system modeled as an autoencoder.

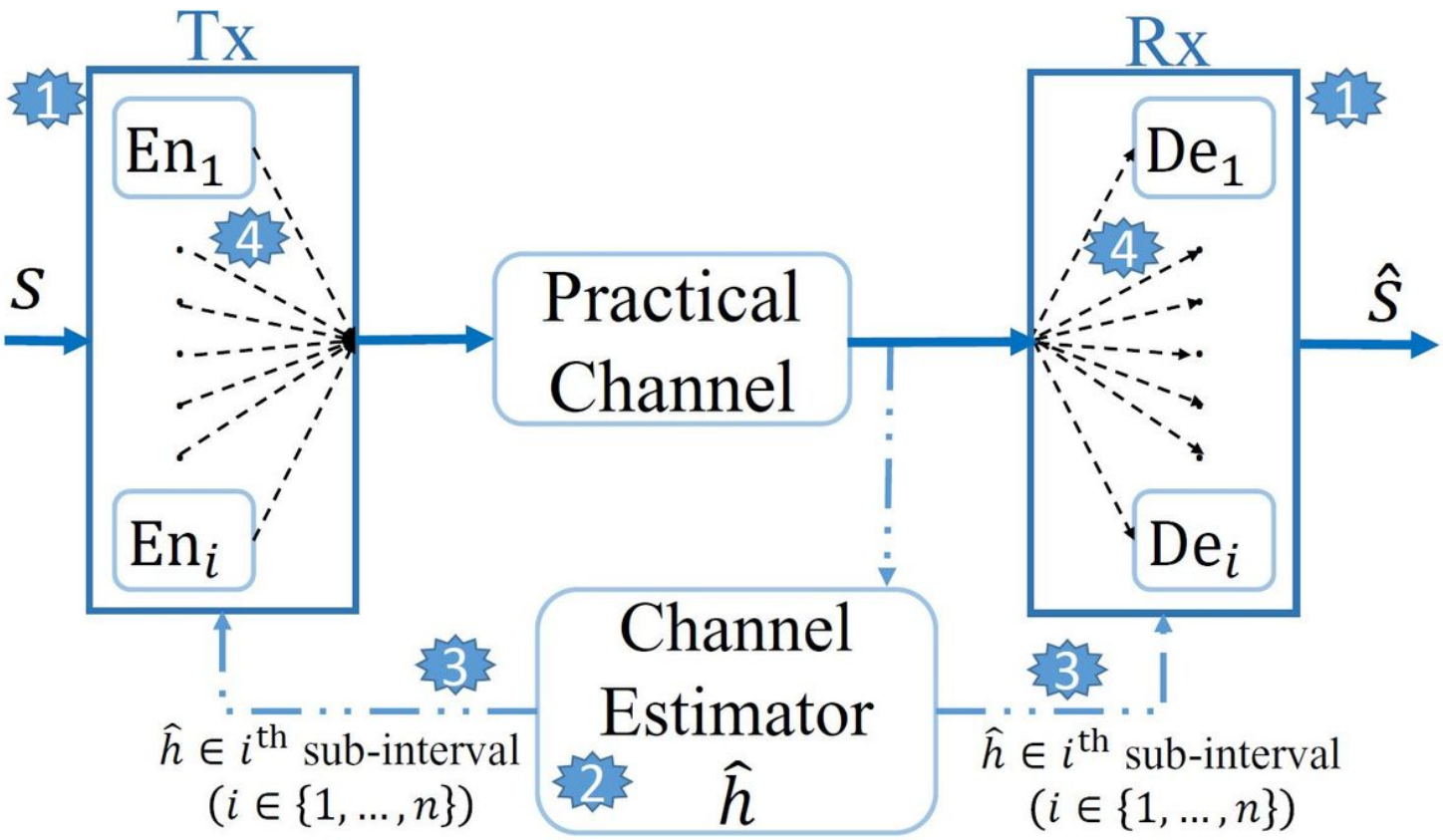

Figure 2

Four-step structure of the proposed adaptive transmission scheme for the DL-based communication system.

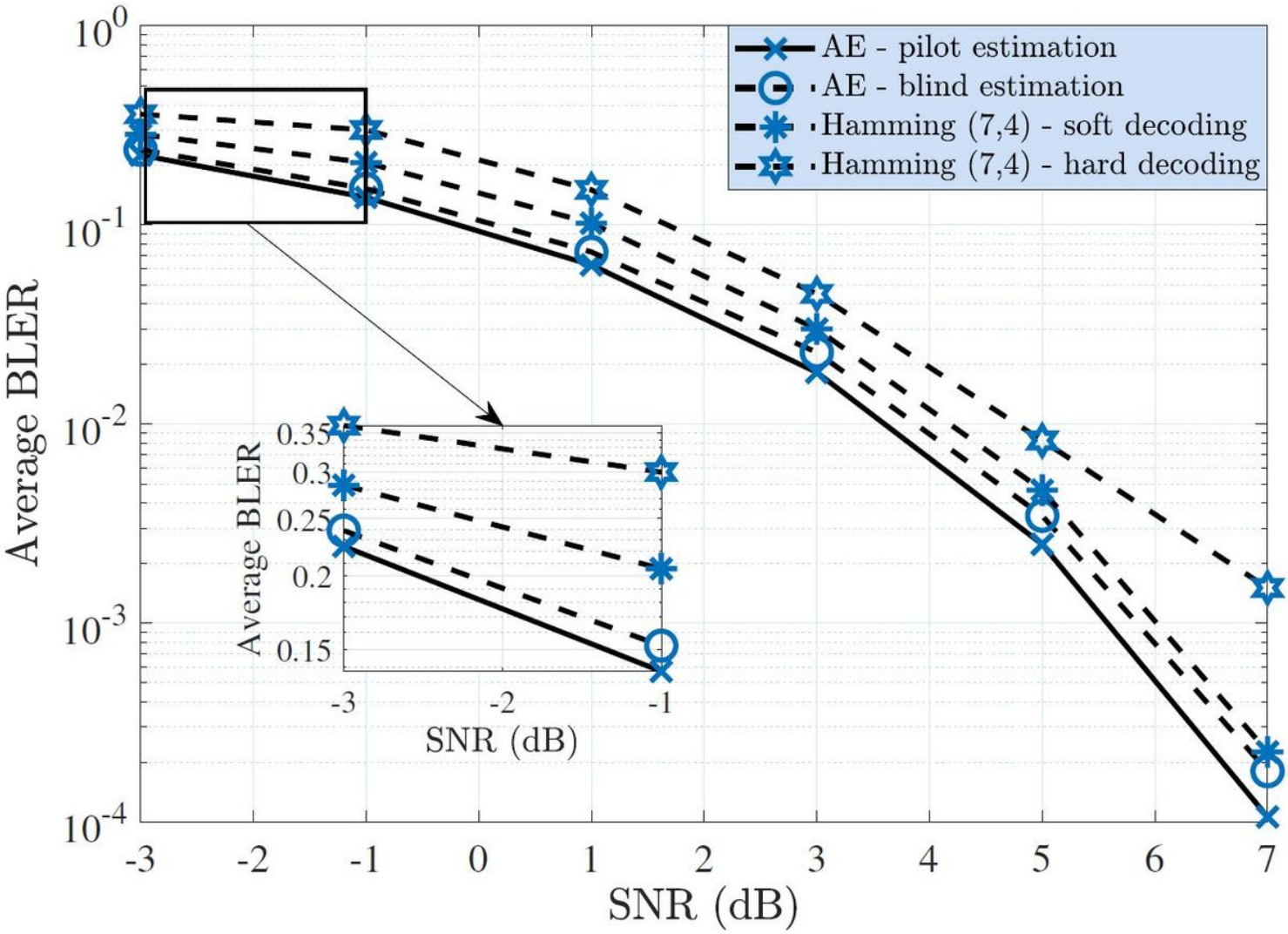

Figure 3

Average BLER versus SNR when the instantaneous channel gain, ^h, is estimated by using two methods, i.e., pilot symbol method, and blind estimation method. The parameters for the AE are set as follows: M = 16, L = 7, and n = 5.

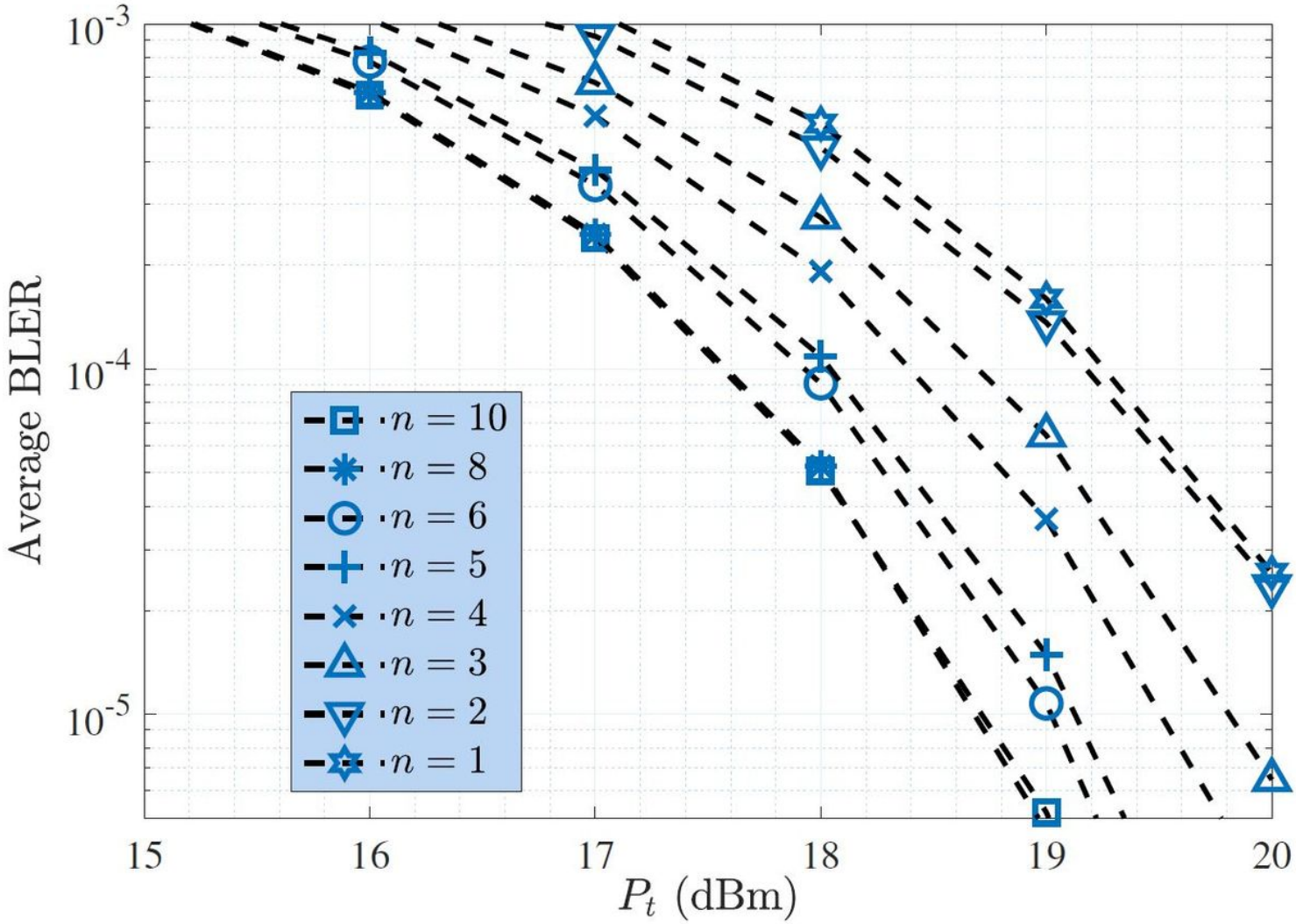

Figure 4

Average BLER versus pt for di erent numbers of sub-intervals n. The parameters for the AE are set as follows: M = 16, L = 7.